

UNITED STATES PATENT APPLICATION

of

JEROME BELLEGARDA

and

WAYNE LOOFBOURROW

for

**METHOD AND APPARATUS FOR AUTOMATIC FILE CLUSTERING
INTO A DATA-DRIVEN, USER-SPECIFIC TAXONOMY**

Attorney Docket No. P2989-908
BURNS DOANE SWECKER & MATHIS, LLP
P.O. Box 1404
Alexandria, Virginia 22313-1404
(703) 836-6620

**METHOD AND APPARATUS FOR AUTOMATIC FILE CLUSTERING
INTO A DATA-DRIVEN, USER-SPECIFIC TAXONOMY**

FIELD OF THE INVENTION

[0001] The present invention relates to the field of graphical user interfaces, and more specifically, to a method of displaying user-generated documents within a file system.

BACKGROUND OF THE INVENTION

[0002] The various files and folders present on a computer system are organized in a complex hierarchy of directories, referred to as the file system. Some of the files and folders within the file system are necessary for the operating system, and the applications it supports, to work properly. These files and folders are logically positioned in the file system, and their organization is well documented for technical support purposes. The remainder of the files are typically created or downloaded by the user in the course of using the computer, and the way they are organized is entirely left up to individual preferences.

[0003] Most users start out with a reasonably principled directory structure, but as time goes by and the complexity of their file hierarchy grows, it typically becomes more and more difficult for them to navigate this ever-expanding portion of the file system. Advanced user interface elements, such as the "column view" in the MacOS X operating system distributed by Apple Computer Inc., are available for them to visualize what the file hierarchy looks like at any given point. In addition, sophisticated search capabilities can help them find the information they want to access, e.g. by file name/characteristics, document content, etc.

[0004] Nevertheless, a far better navigation experience could be achieved if there existed a method for visualizing/displaying documents based on their content, i.e., in a semantic hierarchy. This semantic view option would complement

current directory structures, and likely help users keep their file hierarchies in a readily usable state.

[0005] To make a semantic view possible, it is necessary to classify each user-generated file against a suitable taxonomy, so that files sharing the same taxonomy node can be grouped together accordingly. There are a number of possible approaches to this information management problem.

[0006] A first information management approach is to classify information against an existing all-purpose taxonomy using standard similarity measures. This approach is not particularly adequate, however, because to be useful, the taxonomy needs to be user-specific. For example, consider the concept of "metal." While it connotes a hard material to some users, it represents a type of music for other users. As another example, the term "jaguar" is likely to have a very different meaning to car enthusiasts, to animal lovers, and to personal computer aficionados ("Jaguar" being the code name for the MacOS X v 10.2 operating system).

[0007] A second of the three approaches is to modify the all-purpose taxonomy to more closely reflect the situation at hand, by applying hand-crafted mapping rules. This approach has limitations as well. Setting aside the problem of hand-crafting the mapping rules (a non-trivial endeavor, in itself), typically the method is only able to perform slight modifications on the node labels, not the basic structure of the taxonomy. This may work for some users some of the time, but because it fails to take into account individual preferences, this approach is likely to dilute the perceived value of the result. In the example above, "jaguar" might be very close to the top of the preferred taxonomy for a MacOS X enthusiast, but very deep into it for another person. The ability to re-structure the existing taxonomy to increase the visibility of "jaguar" would probably be critical to the MacOS X enthusiast.

[0008] Finally, the third approach is to first build a user-specific taxonomy by manually defining a set of suitable user-related topics. Classification proceeds by isolating a relatively small, for example 50 to 100, number of documents that are deemed paradigms of each topic, and training a statistical classification system on that data. The statistical classification system is then used to classify the remaining files. This method is clearly not suited to the particular problem at hand, as users are generally not the kind of information specialist capable of laboriously assembling the necessary training sets. Furthermore, as the number of categories increases, this task becomes exponentially more onerous.

SUMMARY OF THE INVENTION

[0009] Accordingly, it is desirable to be able to automatically generate a special purpose taxonomy, revolving around concepts that are not only semantically meaningful but important to the user. Since the only evidence available to construct such a taxonomy is in the set of files to be classified, a satisfactory solution should provide simultaneous training/classification of the files into the user-specific taxonomy.

[0010] The invention overcomes the above-identified problems associated with known classification systems by providing a method and apparatus for hierarchically clustering files and suitably labeling the resulting clusters. In one embodiment of the invention, this is achieved by exploiting a latent semantic analysis (LSA) paradigm, which has proven effective in query-based information retrieval, word clustering, document/topic clustering, large vocabulary language modeling, and semantic inference for voice command and control. More information on latent semantic analysis can be found in the article, "Exploiting Latent Semantic Information in Statistical Language Modeling", by J. R. Bellegarda, *Proc. IEEE*, Vol. 88, No. 8, pp. 1279-1296, August 2000, hereby incorporated by reference.

[0011] In accordance with the invention, the above-mentioned objectives are achieved by incorporation of a semantic view option within the graphical user interface. When invoked, this view employs a clustering and labeling algorithm that results in the creation of semantic hierarchy of all user-generated documents based on document content. Thus, the user is able to navigate among documents based on their content, rather than some other organizational structure.

[0012] Further features of the invention, and the advantages offered thereby, are explained in greater detail hereinafter with reference to specific embodiments illustrated in the accompanying drawings, wherein like elements are designated by like identifiers.

BRIEF DESCRIPTION OF THE DRAWING FIGURES

[0013] The objects and advantages of the invention will be understood by reading the detailed description in conjunction with the drawings, in which:

FIG. 1 shows a block diagram of an exemplary computer system in which the invention can be employed;

FIGS. 2A and 2B show an exemplary conventional file hierarchy and a semantic hierarchy in accordance with the invention, respectively;

FIG. 3 is a flow chart illustrating the creation of a semantic hierarchy in accordance with an exemplary embodiment of the invention;

FIG. 4 illustrates a matrix that is constructed from a set of text documents; and

FIG. 5 depicts the singular valued decomposition of the matrix.

DETAILED DESCRIPTION OF THE INVENTION

[0014] To facilitate an understanding of the principles and features of the invention, it is explained hereinafter with reference to its implementation in an illustrative embodiment. In particular, an example is provided in which text

documents are analyzed and clustered on the basis of their word content. It will be appreciated, however, that the present invention can find utility in a variety of applications to various types of data files, as will become apparent from an understanding of the principles that underscore the invention.

[0015] An exemplary computer system of the type in which the present invention can be employed is illustrated in block diagram form in Figure 1. The structure of the computer itself can be of a conventional type. It is briefly described here for subsequent understanding of the manner in which the features of the invention cooperate with the structure of the computer.

[0016] Referring to Figure 1, the system includes a computer 100 having a variety of external peripheral devices 108 connected thereto. The computer 100 includes a central processing unit 112, a main memory which is typically implemented in the form of a random access memory 118, a static memory that can comprise a read only memory 120, and a permanent storage device, such as a magnetic or optical disk 122. The CPU 112 communicates with each of these forms of memory through an internal bus 114. Additionally, other types of memory devices may be connected to the CPU 112 via the internal bus 114. The peripheral devices 108 include a data entry device such as a keyboard 124, and a pointing or cursor control device 102 such as a mouse, trackball or the like. A display device 104, such as a CRT monitor or an LCD screen, provides a visual display of the information that is being processed within the computer, for example the contents of a document or a hierarchical view of multiple documents and folders. A hard copy of this information can be provided through a printer 106, or similar device. Each of these external peripheral devices communicates with the CPU 112 by means of one or more input/output ports 110 on the computer. Input/output ports 110 also allow computer 100 to interact with a local area network (LAN) server or an external network 128, such as a VLAN, WAN, or the Internet 130.

[0017] Computer 100 typically includes an operating system (OS), which controls the allocation and usage of the hardware resources such as memory, central processing unit time, disk space, and peripheral devices. The operating system includes a user interface that is presented on the display device 104 to enable the user to interact with the functionality of the computer. If the user interface is a graphical user interface (GUI), the operating system controls the ability to generate windows and other graphics on the computer's display device 104. For example, the operating system may provide a number of windows to be displayed on the display device 104 associated with each of the programs running in the computer's RAM 118. Depending upon the operating system, these windows may be displayed in a variety of manners, in accordance with themes associated with the operating system and particular desired display effects associated with the operating system.

[0018] Another component of the operating system is the file system, which controls access to and organizes the files stored in the computer system, such as the local storage disk 122 and/or remote storage media. The user interface provides a capability for a user to view the contents of the file system. For example, a graphical user interface may provide a hierarchical display of files and folders, or directories, as shown in Fig. 2A. The GUI can also provide other view options, such as by list or icon. These types of views typically correspond to a structural organization designed by the user. As discussed above, these known methods of viewing/navigating file system documents can become cumbersome to the user as the number of files increases.

[0019] Accordingly, the invention provides a semantic view option which allows a user to view documents by, for example, the content of the file. This allows the user a choice of, for example, icons, list, file system columns, or semantic hierarchy. As shown in the hierarchical display of Fig. 2B, the user files are displayed in an hierarchical format based on the content of the documents. This is

achieved, according to one embodiment of the invention, by employing a clustering and labeling algorithm that classifies text files based on the word content of the files. In the context of the present invention, the term "text file" is not limited to "pure" text files, e.g. those generated with a text editor program. Rather, it includes any type of file containing textual content that can be retrieved through a suitable text extraction or file translation process, such as files in PDF format, word processor files, and even image files containing text that can be discerned through optical character recognition or the like.

[0020] In addition to clustering and labeling text files based on semantic similarities within their content, the invention can cluster or organize non-text files in accordance with more traditional methods of clustering based on metadata. For example, graphic files can be organized under a label of "pictures" or they can be further organized based on information provided by the user during creation of the file, using rule-based clustering.

[0021] Clustering of the files can be initiated upon selection of a "semantic view" option within the GUI, and/or run periodically in the background. Once the initial analysis of the documents is performed to derive a taxonomy, re-evaluation of the collection is not necessary every time the user adds a document. As a result newly added documents can be classified against the existing taxonomy, and only if the "fit" is outside acceptable parameters is further evaluation and re-classification of the corpus of documents required. However, if preferable, the evaluation and clustering process can be performed upon creation of a new file, or periodically in the background, for example, when the CPU 112 is not in high use.

[0022] The clustering and labeling algorithm for text files comprises three principal stages: (i) mapping all words and documents into an appropriate semantic vector space; (ii) using semantic similarity to cluster the documents at predetermined levels of granularity; and (iii) assigning a meaningful descriptor to each resulting cluster in the space. These three stages are represented in Figure 3

as steps 301, 303 and 305, respectively. Once the documents have been clustered and labeled according to this process, they are displayed to the user, at step 307, in a manner that is based on the resulting clusters, as represented in Figure 2B.

[0023] Various techniques can be employed to accomplish these tasks. For textual documents, a language model is employed to identify the underlying semantics of the files. In a preferred embodiment of the invention, the statistical model provided by the LSA paradigm is used to implement all three of these stages. In general, scattered instances of word-document correlation are mapped into a parsimonious semantic space during the first stage by means of a dimensionality reduction technique provided by LSA. The second stage utilizes LSA document-to-document comparison capabilities to evaluate all potential clusters. LSA word-document comparison capabilities are used in the final stage to determine the words that are most appropriate for each cluster.

[0024] A detailed description of the implementation of these three stages, using the latent semantic analysis paradigm, follows. Let T be the collection of all N user-generated files present at a given time on the user's computer. This collection is flat, in the sense that it does not retain information about the particular directory structure used to organize the files. Also, let $v, |v|=M$, be the list of words and other symbols that occur in T , i.e., the underlying vocabulary.

[0025] First an $(M \times N)$ matrix W , whose entries w_{ij} suitably reflect the extent to which each word $w_i \in v$ appearing in document $d_j \in T$ is constructed. A reasonable expression of w_{ij} is:

$$w_{ij} = (1 - \epsilon_i) \frac{c_{ij}}{n_j} \quad (1)$$

where c_{ij} is the number of times w_i occurs in d_j , n_j is the total number of words present in d_j , and ϵ_i is the normalized entropy of w_i in the corpus T . The global weighting implied by the expression $1 - \epsilon_i$ reflects the fact that two words appearing with the same count in a particular document do not necessarily convey the same

amount of information; this is subordinated to the distribution of the words in the entire collection T.

[0026] The matrix W resulting from this feature extraction is depicted in Figure 4, and defines two vector representations for the words and the documents. Each word w_i is uniquely associated with a row vector of dimension N, and each document d_j is uniquely associated a column vector of dimension M. In a practical implementation with an appreciable number of documents, the vectors w_i and d_j will typically be quite sparse, i.e. a large number of the cell values w_{ij} will be zero. In addition, the dimensions M and N can get to be quite large, and the dimension spaces are distinct from one another. As explained in greater detail in the above-cited publication, these issues can be addressed by performing a matrix decomposition on the matrix W.

[0027] In one embodiment of the invention, a singular value decomposition is carried out. An R-dimensional singular value decomposition (SVD) of W is depicted in Figure 5, and represented as:

$$W = USV^T, \quad (2)$$

where U is the (M x R) left singular matrix with row vectors $u_i (1 \leq i \leq M)$, S is the (R x R) diagonal matrix of singular values $s_1 \geq s_2 \geq \dots \geq s_R > 0$, V is the (N x R) right singular matrix with row vectors $v_j (1 \leq j \leq N)$, $R \ll M, N$ is the order of the decomposition, and T denotes matrix transposition. This rank-R decomposition defines a mapping between: (i) each word and the R-dimensional vector $\bar{u}_i = u_i S$, after appropriate scaling by the singular values, and (ii) each document and the R-dimensional vector $\bar{v}_j = v_j S$, after the same scaling. The continuous vector space spanned by all of the instances of \bar{u}_i and \bar{v}_j is referred to as the LSA space, S.

[0028] To understand the semantic nature of the mapping, it can be observed that the relative position of the R-dimensional vectors is determined by the overall pattern of the language used in T, as opposed to specific keywords or constructs.

Hence a word whose meaning is related to w_i will tend to map to a vector "close" (in some suitable metric) to $\overline{u_i}$, while a document germane to the topic discussed in d_j will tend to map to a vector "close" to $\overline{v_j}$. These characteristics form the basis for clustering and labeling.

[0029] Since the space S is continuous, it is only necessary to identify an appropriate closeness measure to enable document clustering. For the LSA paradigm, a natural metric to consider is the cosine of the angle between two document vectors. Thus a suitable measure for document-to-document comparison is given in equation (3) for $1 \leq j, k \leq N$:

$$K(\overline{v_j}, \overline{v_k}) = \cos(v_j S, v_k S) = \frac{v_j S^T v_k S}{\|v_j S\| \|v_k S\|} \quad (3)$$

Clustering occurs by evaluating which two documents are closest to each other, and merging their semantic information together.

[0030] The number of clusters at any given level of granularity can be controlled by monitoring the increase in cluster variance resulting from a merge operation. Since the underlying singular vectors are orthogonal, covariance matrices are diagonal. Thus, it is sufficient to consider what happens along any one dimension. Along that dimension, let μ_1, σ_1^2 and μ_2, σ_2^2 be the means and variances of two candidates for merging. If n_1 and n_2 are the sizes of the two clusters, the mean variance of the merged entity along that dimension is:

$$\sigma^2 = \frac{n_1 \sigma_1^2 + n_2 \sigma_2^2}{n_1 + n_2} + \frac{n_1}{n_1 + n_2} \cdot \frac{n_2}{n_1 + n_2} \cdot (\mu_1 - \mu_2)^2 \quad (4)$$

Thus, the merge operation is guaranteed to increase the cluster variance over and beyond the average variance of the two candidates, by a quantity seen to be proportional to $(\mu_1 - \mu_2)^2$. This quantity can easily be tracked, and suitable thresholds established to implement any desired level of granularity. Thus, a first

threshold can be defined to establish the lowest level of clusters into which the documents will be grouped, and additional thresholds can define higher level clusters, or "super clusters", in which plural lower-level clusters are grouped. These higher level clusters might also include outlier documents that do not fall within the thresholds to be included in lower-level clusters.

[0031] Once the clusters are derived, they are labeled in a meaningful way for presentation to the user. To do that, the word(s) most representative of the cluster content are determined, which is accomplished by means of a word-document comparison in the LSA space S . In the LSA paradigm, a natural metric to consider is the cosine of the angle between the associated word and document vectors, taking the appropriate scaling into account. Thus a suitable closeness measure for $1 \leq i \leq M$, $1 \leq k \leq N$, is

$$\overline{K}(\overline{u_i}, \overline{v_k}) = \cos(u_i S^{1/2}, v_k S^{1/2}) = \frac{u_i S v_k^T}{\|u_i S^{1/2}\| \|v_k S^{1/2}\|} \quad (5)$$

All words present in the cluster need not be evaluated, since function words, for example, would not be meaningful. On the other hand, there may be words from the underlying vocabulary which do not occur in the cluster, but may still be relevant. It therefore may be advantageous to filter beforehand the pertinent subset of the vocabulary (e.g., nouns, verbs, and adjectives) deemed the most promising to evaluate.

[0032] Applying the metric (5) results in a list of candidate labels for each cluster, ranked in decreasing order of relevance. Those that are within a pre-determined threshold, and optionally satisfy any other suitable criteria (such as further part-of-speech constraints, for example), can be retained. These words constitute the label descriptor returned to the user to characterize the cluster. Repeating this procedure for each cluster at every level of granularity completes the taxonomy sought.

Example

[0033] Preliminary experiments were conducted using a database of 324 files varying in length from 14 to 3328 words, with an average length of 471 words. This sample set is reasonably representative of the range of text document sizes likely to be produced by an average user. The general domain was financial news, which is narrower than the typical user's. Accordingly, this database translates into fairly severe test conditions.

[0034] The approach described above was used to derive a hierarchical structure with 3 levels of granularity. The bottom level (level-3) comprised the 324 documents themselves, the middle level (level-2) a total of 20 clusters, and the top level (level-1) 5 superclusters. No word agglomeration was performed, so label descriptors comprised individual words only. The top 3 or 4 words were retained for the purpose of illustration. In a preferred embodiment, word agglomeration would better capture multi-word expressions like "interest rate."

[0035] Table I offers a partial display of the resulting semantic view for this test set, showing all 5 level-I superclusters but only 8 of the 20 level-2 clusters. When compared to a subjective manual organization, the misclassification error rate at the 20 cluster level was measured to be 6.3 percent. This compares favorably with the typical misclassification rate available in the prior art (from 10% to 15% assuming an existing all-purpose taxonomy with suitably modified labels). In addition, the approach described above has the advantage to build, in a completely autonomous fashion, a taxonomy individually customized to each user.

Level - 1	Level - 2	Level - 3
electronics glut semiconductor shipments	chip difficulties manufacturers	file.0
		file.15
		file.37
	engineering equipment valero	etc...
		file.1
		file.27
		file.30
		etc...
concern credit investment revenues	debenture grant receipts	file.4
		file.8
		file.25
		etc...
	group management panic	file.5
		file.6
		file.12
		etc...
accountability bilateral economy testified	act presidential subcommittee	file.7
		file.35
		file.36
		etc...
	allied command formation	file.11
		file.23
		file.51
		etc...
aids doctors laboratories portfolio	interferon patients patent	file.10
		file.18
		file.19
		etc...
airwaves entertainment hollywood television	broadcasting cable communications	file.2
		file.21
		file.21
		etc...

Table 1

[0036] In accordance with the present invention, once the clustering of documents into a suitable number of levels, and the labeling of the clusters, has been performed, the documents are displayed to the user in a view that

corresponds to the derived taxonomy. An example of such a view, based upon the foregoing example, is depicted in Figure 2B. In a manner analogous to the conventional file system view of Figure 2A, the documents are represented in a hierarchical arrangement of folders and files, where the folders correspond to the respective clusters. In this case, however, the organization of the folders and files is based upon the content of the documents, rather than the file system structure. As a result, when a new document is added to the collection, it can be automatically classified and displayed in the appropriate folder, without user intervention. Furthermore, the view presented to the user is dynamically adaptive to changes in the overall content of the collection as documents are added and subtracted from it.

[0037] The semantic view of the present invention is preferably incorporated into the graphical user interface as one of a number of selectable options from which the user can choose. Thus, a default view might be the hierarchical tree view of Figure 2A, in which the files are organized in accordance with their path names, i.e. the actual file system structure. To facilitate access to a particular file whose location may not be intuitive, the user can switch to the semantic view of Figure 2B, and thereby select it on the basis of its content, rather than its location. As a further option, the semantic hierarchy might be presented in a column view, rather than a tree view. If desired, the user can also be presented with the option of switching the actual file system structure based upon the virtual file system arrangement presented in the semantic view.

[0038] The foregoing embodiment of the invention has been described with reference to its implementation using the LSA paradigm to perform all three of the major stages of mapping the corpus of files into a semantic vector space, clustering the files within the space, and assigning labels to the clusters. While this particular paradigm is preferred for textual documents because it accomplishes the results in a statistically sound manner, it does not represent the sole approach for

achieving the principles of the invention. Rather, any language model which has the ability to capture the underlying semantics of the files can be employed to present the user with a content-based view of the file system. In a simplistic approach, for instance, a thesaurus-based synonym expansion might be used to perform some of the stages. As another possibility, a form of n-gram analysis, incorporating some suitable span extension, might be used.

[0039] It will be appreciated, therefore, that the present invention can be embodied in other specific forms without departing from the spirit or central characteristics thereof. For example, while the invention has been described in the context of clustering text files based on the word content of the files, the invention is equally applicable to the semantic views based on other methods of clustering. For instance with respect to non-textual data files, the clustering can be based upon file metadata and the like.

[0040] Furthermore, provision can be made for the user to override a particular clustering or labeling outcome, with feedback propagated to the semantic space as appropriate. For instance, if the user moves a document from one cluster to another, the relative weighting of words could be adjusted to conform with the new alignment. Similar results can take place if the user changes the label for a cluster.

[0041] The presently disclosed embodiments are, therefore, considered in all respects to be illustrative and not restrictive. The scope of the invention is indicated by the appended claims, rather than the foregoing description, and all changes that come within the meaning and range of equivalents thereof are intended to be embraced therein.